Amendments to the Claims

1. (Original) A method of handling events received at sockets in a computer server configured to serve clients, the method comprising:

executing a polling module configured to poll server sockets to detect events received at said sockets;

registering a first plurality of sockets with said polling module, wherein each of said sockets in said first plurality of sockets is associated with an event consumer;

notifying a first event consumer associated with a first socket in said first plurality of sockets when a first event is received at said first socket; and

invoking a task configured to facilitate handling of said first event;

wherein a first processor thread is shared among said first plurality of sockets for said polling; and

wherein one or more processor threads are allocated to the execution of tasks invoked by said event consumers.

2. (Original) The method of claim 1, further comprising:

registering a second plurality of sockets with said polling module, wherein each of said sockets in said second plurality of sockets is associated with an event consumer;

wherein a second processor thread is shared among said second plurality of sockets.

3. (Original) The method of claim 1, wherein the server is configured to stream media to the clients.

4. (Currently Amended) The method of claim 3, wherein said event consumers are program objects, and each of said event consumers is one of the set of:

a listener consumer configured to handle a connection request event;

a connection consumer configured to handle a media streaming command event; and

a receiver consumer configured to handle a media stream quality event;

wherein said listener consumer, said connection consumer and said receiver consumer are derived from a single abstract base class ~~implementations of a single event consumer interface class~~.

5. (Currently Amended) The method of claim 3, wherein said first event comprises a connection request from a client; and

wherein said first event consumer is a listener event consumer configured to establish a client connection through a second socket in response to said connection ~~request~~ request.

6. (Original) The method of claim 3, wherein said first event comprises a media streaming command; and

wherein said first event consumer is a connection consumer configured to execute said media streaming command.

7. (Original) The method of claim 3, wherein said first event comprises media stream quality information; and

wherein said first event consumer is a receiver consumer configured to adjust said media stream according to said media stream quality information.

8. (Original) The method of claim 3, wherein said first socket is configured to host connections with multiple clients simultaneously.

9. (Original) The method of claim 3, wherein said first event comprises a request for a media streaming control connection from a first client and said first event consumer is a listener event consumer, and wherein said invoking a task comprises:

establishing a media streaming control connection with the first client through a second socket configured for media streaming control connections with multiple clients;

wherein said second socket is associated with a connection event consumer configured to handle a media streaming control command.

10. (Original) The method of claim 9, further comprising:

receiving, from the first client at said second socket, a media streaming command to stream media to the first client;

establishing a media streaming quality connection with the first client through a third socket configured for media streaming quality connections with multiple clients;

wherein said third socket is associated with a receiver event consumer configured to handle media streaming quality information.

11. (Original) A method of handling events received at a server configured to stream media to clients, wherein processor resources within the server are allocated in the form of threads, comprising:

polling one or more registered server sockets for events received at the server from clients, wherein each registered socket is associated with an event consumer configured to handle an event received at the registered socket;

receiving a client connection request at a first socket;

notifying a first event consumer of the connection request, wherein said first event consumer is associated with said first socket;

registering a second socket configured to receive media streaming commands through a connection established in response to said client connection request;

receiving at said second socket a media streaming command from the client;

notifying a second event consumer of the command, wherein said second event consumer is associated with said second socket; and

issuing one or more tasks configured to execute the media streaming command;

wherein a first thread is shared among a first collection of sockets comprising said first socket and said second socket; and

wherein a set of threads is allocated to said one or more tasks.

12. (Original) The method of claim 11, further comprising:

registering a third socket configured to receive data concerning the quality of media being streamed to the client, wherein said third socket is associated with a third event consumer;

wherein said first collection of sockets includes said third socket.

13. (Original) The method of claim 11, wherein said second socket is configured to receive media streaming commands from multiple different clients.

14. (Original) A computer readable storage medium storing instructions that, when executed by a computer, cause the computer to perform a method of handling events received at a server configured to stream media to clients, wherein processor resources within the server are allocated in the form of threads, the method comprising:

polling one or more registered server sockets for events received at the server from clients, wherein each registered socket is associated with an event consumer configured to handle an event received at the registered socket;

receiving a client connection request at a first socket;

notifying a first event consumer of the connection request, wherein said first event consumer is associated with said first socket;

registering a second socket configured to receive media streaming commands through a connection established in response to said client connection request;

receiving at said second socket a media streaming command from the client;

notifying a second event consumer of the command, wherein said second event consumer is associated with said second socket; and

issuing one or more tasks configured to execute the media streaming command;

wherein a first thread is shared among a first collection of sockets comprising said first socket and said second socket; and

wherein a set of threads is allocated to said one or more tasks.

15. (Original) A computer readable storage medium containing a data structure configured for facilitating the handling of events received at communication sockets in a media streaming server, the data structure comprising:

a plurality of socket identifiers, wherein each socket identifier is configured to identify a server socket established to receive a media streaming event; and

for each of said sockets, a reference to an event consumer configured to handle said event by invoking a set of tasks;

wherein a single processor thread is shared among said plurality of sockets for detecting said events; and

wherein a set of processor threads is allotted to the execution of said tasks invoked by said event consumers.

16. (Original) An apparatus for handling media streaming events, comprising:

a polling module configured to poll sockets, wherein said sockets are configured to receive media streaming events from clients;

a polltable comprising a first set of sockets polled by said polling module, wherein a first processor thread is dedicated to said polling of said first set of sockets;

a listener module configured to:

receive a request for a media streaming control connection through a first socket in said first set of sockets; and

establish the requested media streaming control connection through a second socket in said first set of sockets;

a connection module configured to receive a media streaming command through said second socket; and

a third socket configured to stream media.

17. (Original) The apparatus of claim 16, further comprising:

a receiver module configured to receive quality information regarding a media stream.

18. (Original) The apparatus of claim 16, further comprising:

a receiver module configured to receive media, from a media server, for streaming to the client through said third socket.

19. (Original) The apparatus of claim 16, further comprising:

a task queue configured to queue a task invoked by one of said listener module and said connection module;

wherein a set of threads is allocated to execute tasks queued in said task queue.

20. (Original) The apparatus of claim 16, further comprising a task module invoked by said listener module to create said second socket.

21. (Original) The apparatus of claim 16, further comprising a task module invoked by said listener module to create said connection module.

22. (Original) The apparatus of claim 16, further comprising a task module invoked by said connection module to carry out said media streaming command.

23. (Original) The apparatus of claim 16, wherein said listener module and said connection module are program objects generated from a program object class configured to receive media streaming events.